A Study on Object-Oriented Multi-Level Data Model of Public transport

Yun Li
School of Architecture, Chang’an University, Xi’an City, 710061, Shanxi Province, China

Abstract

The data model as the basis of public transportation geographic information system, determine the structure of the data in the system, to ensure the system data storage, editing, maintenance of logic and science, is the key to achieve the function of the system. In this paper, based on the Object-oriented technology, it constructed to meet the multi-level complex data model of geographic information system in transportation demand. In the classification of objects, it divides the objects into spatial object and non-spatial object. In hierarchical structure, it classifies public transportation as semantic level, non-directed level and directed level. In the realization of model, it uses UML and CASE tools in ArcGIS to express public transportation objects and establish object model. This model incorporated all objects into one unified system so as to achieve multi-scale application of the public transportation information system.

Keywords: Object-oriented, UML, Public Transport Object, Data Model.

1. INTRODUCTION

Data model for public transport serves as the basis for the design of public transportation information system. Existing information system for public transport largely use a simplified data representation of the objects such as bus routes and bus stops. In these systems, each bus route is represented by a linear feature, while the direction of the route is left out. Each bus stop is simplified into a point feature and the location of the bus stop is not represented. These data models only enable a general analysis of the spatial layout of public transport objects during public transport planning or the creation of simple information system. But in a real multi-level transit trip system, each bus route has two directions, each of which corresponds to a different timetable. Besides, the geographic location of each bus stop is required for transit trip search, based on which a detailed graphic representation of the bus transfer is drafted. To meet these demands, in detailed transit information system, the expression of normal transit route and transit stop should be refined, which means refined data modeling (Huang, 2003).

There are numerous studies devoted to data model for public transport, ranging from the conventional topological relation models to the present-day object-oriented model. Despite the advances made in integrity, complexity and reliability of the model, some major defects are awaiting resolving. Choi and Jang utilized GIS-based spatial analysis and dynamic segmentation to create an independent network of public transport entities (Choi and Jang, 2000). This network example differentiated between unidirectional and bidirectional bus stops, but it did not make a representation of complex public transport scenarios and directed bus routes. Trepanier and Chapleau applied the object-oriented model to the representation of operation data and road network of public transport (Trepanier and Chapleau, 2001). To accommodate different needs, public transport routes were defined on four levels, and the easiest way of representation was to resort to time points. The most detailed is based on specific route description. This model of
public transport object entities consisted of bus routes, directions of route and bus stops, and the bus route network was correlated to the basic road network via the bus stops. However, details of bus routes were not stored in the database, but it could be reconstructed under GIS. Moreover, although directed routes were represented in this model, no explanations were provided as to its application values. Huang and Peng presented an object-oriented data model for dynamic transportation from the perspective of optimizing the transit trip for users (Huang and Zhong, 2002). By introducing the concept of object, this model considered roads, bus routes, bus stops and time as objects with features. The most salient advantage of the model was its temporal feature, which refers to the fact that the route of the same bus line service at different time was different and was generated dynamically. Because of these features, the model was capable of updating and maintaining the public transport network and outperformed ER models. However, the model was limited in that the bus routes were less refined and the one-way roads were not considered. Since the model applied to data of detailed level, it would be a waste of resources for the general management practice which does not require the data of detailed level. As to the application aspect, some attempts have been made in refining the databases of bus routes and bus stops. For example, the traffic department of Virginia of the United States once used GIS technology for public transportation planning, operational management and propaganda (Jia and Ford, 1999). The locations of the bus stops were collected by GPS and directed representation was done for the bus routes, though a detailed elaboration on these procedures was lacking. In another example, Sarasua et al. provided a detailed description of the procedures of creating a database of bus stops, which included the locations and other information of the bus stops (Sarasua et al., 1998). Refining the database of bus stops is painstaking but a worthwhile effort for the planning and management. The study was devoted to creating a database of bus stops, so no further discussion was carried out on the relationship between bus stops and bus routes or on the multi-mode issue.

Overall, existing data model for public transport display distinctive features in the level of details of entity representation and the correlation to the road network. However, an integrated public transport data model is not yet available. Also lacking is a discussion on the necessity and practicability of data expression of the most detailed level. In other words, and the necessity and practicability of the data expression in the most detailed level is also lack of effective discussion. This study created a multi-level data model for public transport by using the object-oriented technology, which divides the objects into spatial object and non-spatial object. Multiple levels refer to the semantic level, non-direction level and directed level. Unified modeling language (UML) was used to represent the objects, which was followed by model realization based on CASE tool in ArcGIS. After being refined, this data model can be provide scientific and effective basis in the optimal bus route search, travel guide, bus transfer and other aspects of public transport information system establishment and public transport planning.

2. MATERIALS AND METHODS

2.1 UNETRANS (The Unified Network for Transportation) data model

In recent years, with the development and maturity of Object-oriented technology, a new type of Object-oriented data model for transportation is developed, which is the UNETRANS (The Unified Network for Transportation) data model jointly developed by University Consortium for Geographic Information Science (UCGIS) and ESRI. This model is an object-oriented data model that can be used in ArcGIS by any transportation-related application development team. This model is specified in an industry-standard modeling notation called the Unified Modeling Language and is intended as a starting point for structuring transportation data. The physical UML is a detailed view of the design; it provides data types, relationships, and other details. This is the view of the public transport object entities consisted of bus routes, directions of route and bus stops, and the bus route network was correlated to the basic road network via the bus stops. However, details of bus routes were not stored in the database, but it could be reconstructed under GIS. Moreover, although directed routes were represented in this model, no explanations were provided as to its application values. Huang and Peng presented an object-oriented data model for dynamic transportation from the perspective of optimizing the transit trip for users (Huang and Zhong, 2002). By introducing the concept of object, this model considered roads, bus routes, bus stops and time as objects with features. The most salient advantage of the model was its temporal feature, which refers to the fact that the route of the same bus line service at different time was different and was generated dynamically. Because of these features, the model was capable of updating and maintaining the public transport network and outperformed ER models. However, the model was limited in that the bus routes were less refined and the one-way roads were not considered. Since the model applied to data of detailed level, it would be a waste of resources for the general management practice which does not require the data of detailed level. As to the application aspect, some attempts have been made in refining the databases of bus routes and bus stops. For example, the traffic department of Virginia of the United States once used GIS technology for public transportation planning, operational management and propaganda (Jia and Ford, 1999). The locations of the bus stops were collected by GPS and directed representation was done for the bus routes, though a detailed elaboration on these procedures was lacking. In another example, Sarasua et al. provided a detailed description of the procedures of creating a database of bus stops, which included the locations and other information of the bus stops (Sarasua et al., 1998). Refining the database of bus stops is painstaking but a worthwhile effort for the planning and management. The study was devoted to creating a database of bus stops, so no further discussion was carried out on the relationship between bus stops and bus routes or on the multi-mode issue.

Overall, existing data model for public transport display distinctive features in the level of details of entity representation and the correlation to the road network. However, an integrated public transport data model is not yet available. Also lacking is a discussion on the necessity and practicability of data expression of the most detailed level. In other words, and the necessity and practicability of the data expression in the most detailed level is also lack of effective discussion. This study created a multi-level data model for public transport by using the object-oriented technology, which divides the objects into spatial object and non-spatial object. Multiple levels refer to the semantic level, non-direction level and directed level. Unified modeling language (UML) was used to represent the objects, which was followed by model realization based on CASE tool in ArcGIS. After being refined, this data model can be provide scientific and effective basis in the optimal bus route search, travel guide, bus transfer and other aspects of public transport information system establishment and public transport planning.

2. MATERIALS AND METHODS

2.1 UNETRANS (The Unified Network for Transportation) data model

In recent years, with the development and maturity of Object-oriented technology, a new type of Object-oriented data model for transportation is developed, which is the UNETRANS (The Unified Network for Transportation) data model jointly developed by University Consortium for Geographic Information Science (UCGIS) and ESRI. This model is an object-oriented data model that can be used in ArcGIS by any transportation-related application development team. This model is specified in an industry-standard modeling notation called the Unified Modeling Language and is intended as a starting point for structuring transportation data. The physical UML is a detailed view of the design; it provides data types, relationships, and other details. This is the view of the
database design that can be used to create a geodatabase. The specific data needs of any given user will very likely require modifications of, or extensions to, the basic UNETRANS model (Kevin et al., 2003).

UNETRANS data model has the following 6 types of objects and feature packages. Reference network package deals with the objects constituting the geometric and topological networks; routes and location referencing package relates to objects with location referencing information, such as facilities, activities, events and mobile objects; assets package relates to the objects relevant to but not included in the network, such as traffic lights and traffic signs; activities package is concerned with transient objects relevant to but not included in the network, such as planned road maintenance; incidents package is concerned with emergency objects, such as traffic accidents; mobile objects package is concerned with mobile objects relevant to traffic via the network, such as the vehicles. This model addresses the entire field transportation field, so the research in the field of public transport should be made a corresponding choice.

2.2 Object-oriented technology

Object-oriented technology is a program design technology in which objects contain both data and the instructions that work on the data. Centering around data instead of functions, the object-oriented technology can achieve higher stability by encapsulating the data and the instructions together. Thus, class, a new type of data, is created through data abstraction and information hiding, with due consideration given to the connections between different classes and reusability of classes. Class and object are the two cornerstones of the object-oriented technology (Peter and Edward, 1992). Object is made up of attribute and action, and it is meaningful only by having attributes and actions. Attribute is one data item describing the static features of the object. In contrast, action is an operation describing dynamic features of the object. Therefore object is an abstract entity containing features of objective things and a capsule having attributes and actions. In data model design, object is expressed as the sum of data and the operations working on the data. Class is a set of objects sharing common attributes and actions, considered a unified, abstract description of all objects belonging to this class and also having attributes and actions in itself. Class is an abstraction of the set of objects.

2.3 UML

The development of OOA&D (Object-oriented Analysis and Design) method has gained a momentum from late 1980s to the middle of 1990s, and the UML is the product of this period of development. UML (Unified Modeling Language) is generally accepted, which is not only the combination but also the extension of the representing methods such as Booch, Rumbaugh and Jacobson (Meilir, 2001). As a modeling language, the definition of UML includes two parts: UML semantics and UML expression. UML semantics describes the accurate definition of the UML-based meta model. The meta model gives all UML elements with simple, uniform, and universal definitional instructions in terms of syntax and semantics, helping developers to reach a consensus in semantics, and eliminating the influences brought by expressing method that varies from person to person. In addition, UML support the extensional definition of the meta model. UML symbol expression is defined to provide developers or developing tolls with standard usage of symbols and textual syntax for system modeling. What expressed by the symbols and texts belongs to the application-level model, which is an example of the UML meta model from semantic perspective. UML can finish all modeling work that the traditional ER model is able to do, as well as describe some relations that the traditional ER model is unable to do. In addition, UML has many advantages in data base design as compared with tradition ER modeling method: stronger data expression ability; being able to realize the inter-object relations (association and aggregation for instance) which will be
finished via trigger or stored procedure when data base is realized in the future; reserving application program interface for developed data base, and enhancing the robustness, reusability, and portability of system in an unified standard; being directly used for testing at all levels, exposing system designing problem as soon as possible.

3. RESULTS AND DISCUSSION

3.1 Object classification and multi-level representation

An urban public transport system usually has many routes and different bus stops may be distant apart at the same transfer stop. In peak hours, some routes may change or some bus stops may be neglected, giving rise to the situation of “several routes for the same bus line” or “several stops for the same station” (Huang, 2003), as illustrated in Fig. 1. To satisfy different demands of planning, operation, management and service, the objects can be represented on multiple levels, as shown in Fig. 2, by developing a multi-level data model. Therefore, depending on needs, the geodatabase built on the basis of this can shift between detailed or non-detailed levels. As a result, the system overhead can be reduced, while the efficiency and flexibility are increased. Moreover, only the relevant level needs to be revised in times of need rather than the entire data model. This is exactly where the advantages of the object-oriented model lie.

![Figure 1. Schematic of “several stops for the same station”](image1)

![Figure 2. Schematic of public transport objects](image2)

Based on the above, object classification and multi-level structure can be shown in Fig. 3.
Figure 3. The multi-level structure of public transport objects

Spatial objects refer to those having spatial attributes and constituting the bus route network. Non-spatial objects refer to the objects closely related to the public transport and possibly having spatial attributes but not constituting the bus route network. Different objects are associated to each other by aggregation or inheritance, thus composing a complex multi-level structure.

The semantic level only describes the objects and their locations in words, but does not represent the locations with geographic coordinates. The objects belonging to the semantic level are Road, Route, Stop, Vehicle, TimeTable and Incident. For applications not requiring the geographic coordinates, operations are mainly conducted on this level, such as public transport planning and management. The semantic level is the basis for other levels.

On the non-direction level, RouteObject and Stopobject are subdivided into virtual bus route (VirBusRoute) and virtual bus stop (VirBusStop), while the Vehicle is subdivided into bus (Bus) and bus location (BusLocation). TimeTable object is subdivided into route timetable (DirBusRouteTimeTable), stop timetable (DirBusRouteStopTimeTable) and segment timetable (DirBusRouteSegTimeTable). Incident object is subdivided into traffic accident (TrafficAccident) and route closure (RouteClosure). VirBusStop is a set of directed bus stops, and when it is deleted, all included bus stops are deleted as well. If one directed bus stop belongs to two virtual bus stops at the same time, it must not be deleted. VirBusRoute is a simplified set of all bus routes. Since the same bus line has two opposite directions and each direction corresponds to a different route, VirBusRoute consists of all routes in both two directions. Bus contains the route ID, the type of vehicles, passenger capacity, speed and other information. BusLocation involves the geographic location and route segment information related to the current vehicle. DirBusRouteTimeTable describes the start and end time of the bus route. DirBusRouteStopTimeTable describes the waiting time at each stop on each route. DirBusRouteSegTimeTable describes the time spent on each segment of each route. TrafficAccident and RouteClosure are concerned with emergencies happening in the bus network, including the start and end time of emergencies and the route segment where the emergencies take place. Non-direction level is a concretization of the semantic level.
in the geographic space, and it is incorporated into the general traffic information system used for planning, management and information service.

The directed level is a description of the location of the spatial objects, the most detailed level. On this level, virtual routes are directed routes containing location information, which are composed of several route segments. Directed bus stop (DirBusStop) contains the location of the bus stop for a specific route. One station may correspond to several bus stops at different locations. Therefore, DirBusStop is represented by ID and name of the directed bus stop (two separately located bus stops may have the same names but different ID), to which virtual stop it belongs (including ID of the virtual stop) and to which directed route it belongs (including ID of the directed route). Directed bus stops and directed bus route segments (DirBusRouteSeg) make up a geometric network. DirBusRouteSeg is formed by the line linking two adjacent directed bus stops, and the relative sequence of the start and end points of the route determines the direction of the route segment. All route segments connected together form the directed route. For each directed route, the start and end points must be directed stops. In addition, the type of route segments should be noted for the directed route segment. For example, “bus” denotes the ordinary line segment, and “cross” the intersection. Directed bus route (DirBusRoute) is a collective term for the same bus line in two opposite directions, which correspond to two directed routes. The direction information is crucial for the directed route. The directed level, the most detailed level, is the precondition for providing transit trip guidance and optimizing the bus routes. But the directed level is premised on the above two levels and does not exist independently.

Different public transport objects are associated to each other in the following manners: DirBusRouteSeg and DirBusStop forming the geometric network (GeometricNetwork), DirBusRouteSeg composing the DirBusRoute, DirBusRoute associating with DirBusStop by stop sequence (StopSequence), DirBusRoute composing VirBusRoute, DirBusStop composing VirBusStop, DirBusRoute having DirBusRouteTimeTable, DirBusStop having DirBusRouteStopTimeTable and DirBusRouteSeg having DirBusRouteSegTimeTable.

Stops and routes as the spatial objects can be represented either non-independently or independently. Independent representation uses geographic coordinates to denote the locations of stops and routes. Non-independent representation correlates public transport entities to the road network without an explicit record of geographic coordinates of entities. Although the latter presents the associations between the two entities, some treatments of converting non-graphical associations aided by special programming are needed when inquiring for spatial features of the public transport entities. In this paper, considering the spatial features of public transport objects, the stops and routes and other spatial objects are used in an independent way of expression is more appropriate. In fact, the two representation approaches are mutually convertible.

3.2 Object-oriented realization of the multi-level data model

3.2.1 UML representation of public transport objects

(1) Semantic level

The semantic level only describes the objects and their locations in words, without the use of geographic coordinates. So no instantiation is provided here, and the objects are only represented by abstract classes, as shown in Fig. 4.
Figure 4. Class diagram of the semantic level

(2) Non-direction level and Directed level

Non-direction level describes the spatial objects by introducing the geographic coordinates on the basis of the semantic level. This is a refined representation of the non-spatial objects by recording the locations. UML representations of the classes are given in Fig. 5, 6, 7, 8 and 9.

Figure 5. Class diagrams of VirBusStop, VirBusRoute and DirBusRoute

Fig. 5 shows the attributes of the objects. VirBusStop contain ID of themselves (VirBusStopID) and ID (DirBusStopID) of the directed stops belonging to them. Virtual routes contain ID of themselves (VirBusRouteID) and ID of the start and end virtual stops on the route. Directed routes contain ID of themselves (DirBusRouteID) and ID of the start and end directed stops on the route, ID of the route segments composing the route (DirBusRouteSegID) as well as ID of the directed stops on the route.

Figure 6. Class diagrams of TrafficAccident and RouteClosure
In Fig. 6, TrafficAccident and RouteClosure are the subtypes of abstract class Incident. The two only differ in the form of occurrence, and their attributes are inherited from the Incident class. The major attributes are ID of the incident (IncidentID) and start and end time of the incident.

Figure 7. Class diagrams of DirBusRouteSeg and DirBusStop

As shown in Fig. 7, the major attributes of DirBusRouteSeg are ID of the route segments (DirBusRouteSegID), ID of the start and end directed stops (FromDirBusStopID, ToDirBusStopID) on the route segment, type of route segment (DirBusRoteSegType) and ID of the directed route (DirBusRouteID) to which it belongs. The major attributes of the DirBusStop are ID of the directed stops (DirBusStopID), ID of the virtual stops (VirBusStopID) belonging to it and ID of all directed route (DirBusRouteID) to which it belongs.

Figure 8. Class diagrams of DirBusRouteStopTimeTable, DirBusRouteSegTimeTable and DirBusRouteTimeTable

In Fig. 8, the major attributes of DirBusRouteStopTimeTable are ID of the directed routes (DirBusRouteID), ID of the directed stops (DirBusStopID) and waiting time (WaitingTime) at the stops. The major attributes of DirBusRouteSegTimeTable are ID of the directed route segment (DirBusRouteSegID) and driving time over the route segment (DrivingTime). The major attributes of DirBusRouteTimeTable are ID of the route (DirBusRouteID) and the start and end time (RouteStartTime, RouteEndTime) of the route.

Figure 9. Class diagrams of Bus and BusLocation
In Fig. 9, the major attributes of Bus contain ID of the bus (BusID), ID of the route (DirBusRouteID), and passenger capacity (Capacity). The major attributes of BusLocation contain ID of location (LocationID), ID of the bus (BusID), geographic coordinates (NominalXLocation, NominalYLocation) and ID of the route segment (DirBusRouteSegID) to which it belongs.

3.2.2 Realizing multi-level data model based on UML static modeling mechanism

A conceptual multi-level data model for public transport is provided in Fig. 10 based on the analysis of public transport objects.

![Diagram of multi-level data model for public transport](image)

**Figure 10.** The conceptual multi-level data model for public transport

Fig. 10 shows the conceptual multi-level data model for public transport thus built. Lines connecting the adjacent directed stops compose the directed route segments, whose direction is determined by the relative sequence of the start and end points of the route. Route segment type is another attribute of the directed route segment. Two directed routes compose one virtual route. Each directed route has a directed route timetable. Any associated directed stops compose the virtual stops and the directed stops have directed stop timetables. Directed stops and directed route segments compose a geometric network, which is the bus route network. Directed routes are associated to directed stops via StopSequence.

The relationship between objects in UML can be divided into two categories: structure relationship and association relationship (Zhao et al., 2001). Correlation, inheritance and aggregation belong to the former. Association relationship has three forms, one to one,
one to many, and many to many, depending on the real-world relationships between the objects. In this paper, based on the static modeling mechanism of UML, the Microsoft Visio is used to generate the static structure diagram as shown in Fig. 11, 12, 13, 14, 15, 16 and 17.

**Figure 11.** Static structure diagram of the semantic level

In Fig. 11, Incident, TimeTable and Vehicle are non-spatial objects, which are inheritance from ESRI Object class. Route and Stop are spatial objects, which are inheritance from ESRI Feature class.

**Figure 12.** Static structure diagram of the GeometricNetwork

Fig. 12 shows the GeometricNetwork formed by DirBusRouteSeg and DirBusStop, it is modeled with a UML class stereotyped as GeometricNetwork. The relationship of DirBusRouteSeg to GeometricNetwork and DirBusStop to GeometricNetwork is a many-to-many association relationship. DirBusStop and DirBusRouteSeg, considered spatial objects in this study, do not contain complex relationships, and they form a GeometricNetwork, so they are inheritance from ESRI SimpleEdgeFeature class and SimpleJunctionFeature class, respectively. DirBusRouteSeg and DirBusStop have an inheritance relationship with SimpleEdgeFeature class and SimpleJunctionFeature class, respectively. These classes become Feature Class when being converted to the GeoDatabase.
Figure 13. Static structure diagram of Incident

Fig. 13 shows the inheritance relationship between incident classes. IncidentPoint class and IncidentLine class are the subtypes of incident, with TrafficAccident belonging to IncidentPoint and RouteClosure belonging to IncidentLine. TrafficAccident and RouteClosure are subtypes of incident and represented by an association stereotyped as "<<Subtype>>". Conversion of these classes to the GeoDatabase resulted in Table.

Figure 14. Static structure diagrams of VirBusRoute, VirBusStop and DirBusRoute

Fig. 14 shows that the DirBusRoute and VirBusRoute are inheritance from the Route class on the semantic level. VirBusStop are inheritance from the Stop class on the semantic level. HasZ=True and GeometryType=esriGeometryPolyline are tagged values of each class, denoting the additive attribute. These classes become Feature Class when being converted to the GeoDatabase.
Fig. 15 and 16 show that DirBusRouteStopTimeTable, DirBusRouteSegTimeTable and DirBusRouteTimeTable are inheritance from the TimeTable class of the semantic level. Bus and BusLocation are inheritance from the Vehicle class of the semantic level. Conversion of these classes to the GeoDatabase produces a table.

Fig. 17 shows the complex relationships between the classes, which come in three types, namely, Simple Relationship, Composite Relationship and Attributed Relationship. Simple Relationship refers to the one-to-many, one-to-one and many-to-many association relationship, represented simply by association. Composite Relationship only occurs to one-to-many association, represented by aggregation. Attributed Relationship is a many-to-many association, represented by association combined with class stereotyped as<<RelationshipClass>>. For example, one VirBusStop is made up of several DirBusStop. When this VirBusStop is deleted, the DirBusStop belonging to it are deleted as well. So the two have a composite relationship. However, since one DirBusStop can only belong to one VirBusStop, VirBusStop and DirBusStop have a one-to-many relationship. For Attributed Relationship, to store the Primary Key and Foreign Key of Origin Class and Destination Class, new tables are added. In UML programming, they are represented by class stereotyped as<<RelationshipClass>>, with the class name being identical to the name of the association. Moreover, OriginPrimaryKey, OriginForeignKey, DestinationPrimaryKey and DestinationForeignKey are assigned in the tagged values of associations. Fig. 17 shows the relevant keys.
3.3 Verification experiment

3.3.1 Creating the GeoDatabase based on the multi-level data model for public transport

The data model designed with Microsoft Visio can automatically generate database framework under the application system. To achieve this, Microsoft Visio first generated the repository based on the data model built above, and then CASE tool in ArcGIS was used to create the database structure, as shown in Fig. 18.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Features</td>
<td>File Geodatabase Feature Class</td>
</tr>
<tr>
<td>TransportationNetwork</td>
<td>File Geodatabase Feature Class</td>
</tr>
<tr>
<td>Bus</td>
<td>File Geodatabase Table</td>
</tr>
<tr>
<td>BusLocation</td>
<td>File Geodatabase Table</td>
</tr>
<tr>
<td>DirBusRouteSegTimeTable</td>
<td>File Geodatabase Table</td>
</tr>
<tr>
<td>DirBusRouteStopTimeTable</td>
<td>File Geodatabase Table</td>
</tr>
<tr>
<td>DirBusRouteTimeTable</td>
<td>File Geodatabase Table</td>
</tr>
<tr>
<td>IncidentLine</td>
<td>File Geodatabase Table</td>
</tr>
<tr>
<td>IncidentPoint</td>
<td>File Geodatabase Table</td>
</tr>
<tr>
<td>OneRouteHasOneTimeTable</td>
<td>File Geodatabase Relationship Class</td>
</tr>
<tr>
<td>OneSegHaveManyTimeTable</td>
<td>File Geodatabase Relationship Class</td>
</tr>
<tr>
<td>OneStopHaveManyTimeTable</td>
<td>File Geodatabase Relationship Class</td>
</tr>
<tr>
<td>OneVirHaveManyDirStops</td>
<td>File Geodatabase Relationship Class</td>
</tr>
<tr>
<td>RouteComposedBySeg</td>
<td>File Geodatabase Relationship Class</td>
</tr>
<tr>
<td>StopSequence</td>
<td>File Geodatabase Relationship Class</td>
</tr>
</tbody>
</table>

Figure 17. Static structure diagram of complex relationship

Figure 18. GeoDatabase structure in ArcCatalog
The package stereotyped as <<FeatureDataset>> in UML was converted into Feature Dataset in the GeoDatabase and all relationships were converted into Relationship Class. The conversion of non-spatial objects into the GeoDatabase produced a table. The attributes of both spatial and non-spatial objects were treated as attribute data and connected to the objects with keys.

3.3.2 Data input and testing

Public transport data of Xi’an city were input into the GeoDatabase to obtain a prototype database containing the experimental data. This database included all spatial and non-spatial objects of the three levels, as shown in Fig. 19.

![Figure19. DataView of the GeoDatabase](image)

All relationships built by UML existed as tables. After data input, data would be automatically generated in the table, as shown in Fig. 20 and 21.

![Figure20. RouteComposedBySeg table](image)
Stop and Route were selected for verification experiment using the Identify tool in ArcMap, as shown in Fig. 22. It can be seen that VirBusStop with ObjectID of 12 had two DirBusStop, whose ObjectID were 29 and 30, respectively. All DirBusRoute, DirBusRouteSegTimeTable, DirBusStop and VirBusRoute to which the DirBusRoute belongs were generated as well. This can greatly facilitate the functions of bus transfer guidance, display and inquiry.

When data editing is required, such as adding (deleting) directed (virtual) stops, there would be corresponding changes in the associated stops, routes, route segments and attribute tables. For example, when deleting VirBusStop with ObjectID of 12, all associated DirBusStop were deleted as well. At the same time, the record of these stops in StopSequence table and DirBusRouteStopTimeTable were also deleted. If a certain DirBusStop also belonged to other VirBusStop, this stop would not be deleted. When adding DirBusStop, ID of the newly added route segments belonging to the DirBusStop and that of the relevant route were also added. Besides, relevant information of this...
DirBusStop was added manually in the DirBusRouteStopTimeTable, which led to an automatic addition of relevant information of this DirBusStop in the StopSequence table.

After the establishment of the database, the functions of finding the shortest path and bus transfer search were realized by using ArcObject and ArcGIS Engine, and the thematic maps generated could be used for public transport planning. By allowing for dynamic choice of data level for the planning, system resources were greatly saved and data maintenance was made easier.

4. CONCLUSION

In this paper, the traditional public transport objects classification and expression is further described in detail. To accommodate the real-time, dynamic and multi-level features, it divided the public transport objects into spatial and non-spatial objects. Multiple levels were introduced, namely, semantic level, non-direction level and directed level. Incident and TimeTable were taken as non-spatial objects. Thus a multi-level data model for public transport was built by using the Object-oriented technology. Hierarchical refinement of public transport object made the expression of the object structure of public transport is more clear and precise, from the logic level to achieve the mutual connection of public transport object, provides a more reliable and convenient data structure for dynamic database design and use in the physical level.

The verification experiment indicated that the data model could reflect the objects and relationships between the objects accurately. However, in a complex public transport system, there are many other influence factors at work, such as the type of bus vehicles and change of route due to traffic jam caused by road maintenance. Therefore, the dynamic and real-time features of a public transport system should be considered. Further investigations will be focused on the following aspects:

(1) Enhancing dynamic representation to realize dynamic operation and reasoning;

(2) Incorporating other factors of public transport to generate a more comprehensive data framework;

(3) Verifying the validity and extensibility of the data model through applications in the public transport field (e.g., public transport planning, bus transfer guidance and optimizing the transit trip).

5. ACKNOWLEDGMENTS

(1) This study was sponsored by the project of “a multi-level data model for public transport in Xi’an City based on UML” under the 2014 Science and Technology Project Plan of Ministry of Housing and Urban-Rural Development (Project No. 2014-K5-012)

(2) This study was sponsored by the project of “a multi-level data model for public transport stop in Xi’an City based on UML” under the 2016 Social Development and Science and Technology project of Shaanxi Province (Project No.2016SF-430).

6. REFERENCES


Kevin C., Mike G., Valerian N., Steve G. (2003). ArcGIS Transportation Data Model (UNETRANS), ESRI.


